

```
XML_Error tinyxml2::XMLDocument::LoadFile(const char* filename)
```

Load an XML file from disk. Returns XML\_NO\_ERROR (0) on success, or an errorID.

```
const XML_Element* tinyxml2::XMLNode::  
    FirstChildElement(const char* value = 0) const
```

Get the first child element, or optionally the first child element with the specified name (value) of this node.

```
const XML_Element* tinyxml2::XMLNode::  
    LastChildElement(const char* value = 0) const
```

Get the last child element or optionally the last child element with the specified name of this node.

```
const XML_Element* tinyxml2::XMLNode::  
    NextSiblingElement(const char* value=0) const
```

Get the next (right) sibling element, or optionally the next sibling element with the specified name (value) of this node.

```
const char* tinyxml2::XMLElement::GetText() const
```

Convenience function for easy access to the text inside an element.

This is a convenient method for getting the text of simple contained text:

```
<foo>This is text</foo>
```

```
const string str = fooElement->GetText();
```

str will have the value "This is text".

```
const char* tinyxml2::XMLElement::  
    Attribute(const char* name, const char* value = 0) const
```

Given an attribute name, Attribute() returns the value for the attribute of that name, or null if none exists.

For example: const char\* value = ele->Attribute("foo");

The value parameter is normally null. However, if specified, the attribute will only be returned if the name and value match.

This allow you to write code: if (ele->Attribute("foo", "bar")) callFooIsBar();

```
const XML_Attribute* tinyxml2::XMLElement::FirstAttribute() const
```

Return the first attribute in the list for this element.

```
const XML_Attribute* tinyxml2::XML_Attribute::Next() const
```

The next attribute in the list.